[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 12/14
G06F 12/16

[12] 发明专利申请公开说明书

[21] 申请号 03153387.6

[43] 公开日 2005 年 2 月 16 日

[11] 公开号 CN 1581109A

[22] 申请日 2003.8.12 [21] 申请号 03153387.6

[71] 申请人 联想（北京）有限公司

地址 100085 北京市海淀区上地信息产业基地创业路 6 号

[72] 发明人 张 鹏 范晓炬

[74] 专利代理机构 北京德琦知识产权代理有限公司

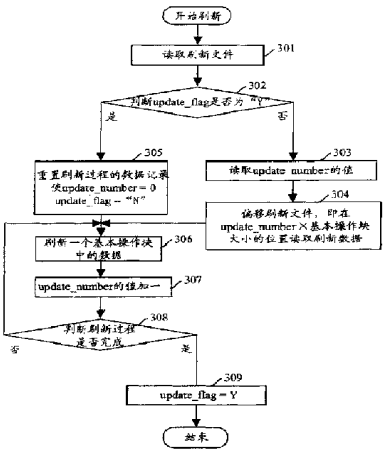
代理人 王 琦

权利要求书 2 页 说明书 5 页 附图 2 页

[54] 发明名称 一种刷新嵌入式系统中非易失性存储器的方法

[57] 摘要

本发明提供了一种刷新嵌入式系统中非易失性存储器的方法，通过将非易失性存储器划分逻辑块，并对每个逻辑块进行编号后，借助保存在硬件上的标志，系统每刷新完一逻辑块都做一次记录，使得系统即使在刷新过程意外中断，也可在下次刷新操作中从中断位置开始，因而在一定程度上减少了升级时间，同时由于不需要对已经成功刷新的部分进行重复的数据写入，从而减少了对非易失性存储器上许多存储单元的刷新次数，降低了因刷新次数过多而引起坏块的危险，同时还减少了由于刷新过程较长而导致的发生意外的几率。应用本发明，不需要增加用户成本，对于强调安全快速的进行刷新的嵌入式系统提供了一种低价可靠的解决方案。



ISSN1008-4274

1、一种刷新嵌入式系统中非易失性存储器的方法，其特征在于，该方法包括以下步骤：

a、将嵌入式系统中的非易失性存储器划分为一个以上的逻辑块，对每个逻辑块编号，并应用其中一逻辑块保存刷新过程中已刷新成功的逻辑块的编号；

b、系统对非易失性存储器进行刷新操作时，读入刷新文件，并判断上次刷新过程是否正常结束，如果是，则将已刷新的逻辑块数的值清零，并执行步骤c，否则获取上次中断的位置，并将已读入的刷新文件定位到相应的位置后，执行步骤c；

c、每刷新完一个逻辑块后，将已刷新逻辑块编号的值加一，并判断刷新过程是否完成，如果是，则结束本次刷新过程，否则，重复执行步骤c。

2、根据权利要求1所述的方法，其特征在于，步骤a所述一个以上的逻辑块大小相同。

3、根据权利要求1所述的方法，其特征在于，步骤a所述保存刷新过程中已刷新成功的逻辑块编号的逻辑块是非易失性存储器中物理位置上的最后一个。

4、根据权利要求1所述的方法，其特征在于，

步骤a所述保存刷新过程中已刷新成功的逻辑块编号的逻辑块中还包括刷新完毕标志；

所述步骤b进一步包括：将已刷新的逻辑块编号的值清零后，将刷新完毕标志置为否，再执行后续步骤；

所述步骤c进一步包括：刷新过程完成后，将刷新完毕标志置为是，在执行后续步骤。

5、据权利要求4述的方法，其特征在于，所述步骤b是根据刷新完毕标志来判断的。

6、据权利要求1所述的方法，其特征在于，所述步骤b是根据已刷新的逻

辑块数乘以逻辑块的大小后所得到的值，获取上次中断的位置。

一种刷新嵌入式系统中非易失性存储器的方法

技术领域

本发明涉及嵌入式系统技术领域，特别是指一种刷新嵌入式系统中非易失性存储器的方法。

背景技术

由于某种原因或嵌入式系统需要升级时，要对保存数据的非易失性存储器进行刷新。现有的刷新流程如图 1 所示。

步骤 101，系统读入刷新所需的镜像文件；

步骤 102，将嵌入式系统中的非易失性存储器上待刷新位置的原有的数据全部擦除；

步骤 103，将镜像文件写入到非易失性存储器上，以达到刷新或升级的目的。

上述方法的缺陷在于：由于嵌入式系统本身用来保存程序和数据的非易失性存储器的刷新速度较慢，刷新过程所需时间相对较长，所以，一旦发生意外断电等使刷新过程中断的情况，将造成系统无法启动，此时必须使用特殊的工具和方法再次对该嵌入式系统中的非易失性存储器重新进行刷新操作，以保证该系统可用。而嵌入式系统中的非易失性存储器的写入次数，即刷新次数通常是受限制的。可见，上述刷新方案即浪费时间又减少了该嵌入式系统的可维护性次数。

发明内容

有鉴于此，本发明的目的在于提供一种刷新嵌入式系统中非易失性存储器的方法，使对嵌入式系统中非易失性存储器刷新操作不但可以从头开始，

也可以从上次刷新的断点处开始。

为达到上述目的，本发明的技术方案是这样实现的：

一种刷新嵌入式系统中非易失性存储器的方法，该方法包括以下步骤：

a、将嵌入式系统中的非易失性存储器划分为一个以上的逻辑块，对每个逻辑块编号，并应用其中一逻辑块保存刷新过程中已刷新成功的逻辑块的编号；

b、系统对非易失性存储器进行刷新操作时，读入刷新文件，并判断上次刷新过程是否正常结束，如果是，则将已刷新的逻辑块数的值清零，并执行步骤 c，否则获取上次中断的位置，并将已读入的刷新文件定位到相应的位置后，执行步骤 c；

c、每刷新完一个逻辑块后，将已刷新逻辑块编号的值加一，并判断刷新过程是否完成，如果是，则结束本次刷新过程，否则，重复执行步骤 c。

较佳地，步骤 a 所述一个以上的逻辑块大小相同。

较佳地，步骤 a 所述保存刷新过程中已刷新成功的逻辑块编号的逻辑块是非易失性存储器中物理位置上的最后一个。

较佳地，步骤 a 所述保存刷新过程中已刷新成功的逻辑块编号的逻辑块中还包括刷新完毕标志；所述步骤 b 进一步包括：将已刷新的逻辑块编号的值清零后，将刷新完毕标志置为否，再执行后续步骤；所述步骤 c 进一步包括：刷新过程完成后，将刷新完毕标志置为是，在执行后续步骤。

较佳地，所述步骤 b 是根据刷新完毕标志来判断的。

较佳地，所述步骤 b 是根据已刷新的逻辑块数乘以逻辑块的大小后所得到的值，获取上次中断的位置。

应用本发明，通过将非易失性存储器划分逻辑块，并对每个逻辑块进行编号后，借助保存在硬件上的标志，使每刷新完一逻辑块都做一次记录，使得系统即使在刷新过程意外中断，也可在下次刷新操作中从中断位置开始，因而在一定程度上减少了升级时间，同时由于不需要对已经成功刷新的部分

进行重复的数据写入，从而减少了对非易失性存储器上许多存储单元的刷新次数，降低了因刷新次数过多而引起坏快的危险，同时还减少了由于刷新过程较长而导致的发生意外的几率，并且减少了再次刷新时的操作时间。应用本发明，不需要增加用户成本，对于强调安全快速的进行刷新的嵌入式系统提供了一种低价可靠的解决方案。

附图说明

图 1 为现有技术的刷新非易失性存储器操作流程示意图；

图 2 为应用本发明的刷新非易失性存储器操作流程示意图；

图 3 为应用本发明的对非易失性存储器实现断点刷新的流程图。

具体实施方式

为使本发明的目的、技术方案和效果更加清楚，以下结合附图及实施例对本发明再做进一步详细的说明。

本发明的思路是：将嵌入式系统中用于保存数据的非易失性存储器划分为一个以上的逻辑块，对每个逻辑块编号，并应用其中一逻辑块保存刷新过程数据记录。图 2 所示为应用本发明的刷新非易失性存储器操作流程示意图。

步骤 201，系统读入刷新所需的镜像文件；

步骤 202，根据保存的升级过程数据记录，获取待刷新位置；

步骤 203，擦除非易失性存储器上一个逻辑块中的数据；

步骤 204，将已擦除数据的逻辑块中写入相应的刷新数据，并返回步骤 202。直到刷新完毕结束。

上述刷新过程数据记录（update_record）中包括用于记录当前已刷新成功的非易失性存储器上的物理位置逻辑顺序的刷新编号（update_number），和用于记录当前刷新过程正常结束的刷新完毕标志（update_flag）。

当系统对非易失性存储器进行刷新操作时，如果上次刷新过程正常结

束，则将 update_number 的值清零，将 update_flag 置为否后，读入刷新文件，每刷新完一个逻辑块，将 update_number 的值加一，当全部刷新完毕后，将 update_flag 置为是后结束本次刷新过程；如果上次刷新过程是非正常结束，则获取上次中断的位置，并将已读入的刷新文件定位到相应的位置后，每刷新完一个逻辑块，将 update_number 的值加一，当全部刷新完毕后，将 update_flag 置为是后结束本次刷新过程。

图 3 所示为应用本发明的对非易失性存储器实现断点刷新的流程图。

步骤 301，读取刷新数据；

步骤 302，判断刷新过程数据记录中的 update_flag 是否为“Y”，如果是，则执行步骤 305，否则执行步骤 303；

步骤 303，读取刷新过程数据记录中的 update_number，以确定上次刷新中断时已刷新过的逻辑块数；

步骤 304，获取上次刷新中断的位置，即用 update_number 的值乘以逻辑块的大小得到上次刷新中断的位置，并将刷新文件偏移到断点处后，执行步骤 306；

步骤 305，重置刷新过程数据记录，即将 update_number 置为零，将 update_flag 置为“N”；

步骤 306，刷新一个逻辑块，即擦除一个逻辑块中的数据内容后，将已擦除数据的逻辑块中写入相应的刷新数据；

步骤 307，update_number 的值加一；

步骤 308，判断升级过程是否完成，如果是，则执行步骤 309，否则返回步骤 306；

步骤 309，将 update_flag 置为“Y”后结束本次刷新过程。

现以嵌入式 Linux 系统为例，具体说明本方案的实现过程：

本实施例的嵌入式 Linux 系统的系统文件被压缩后为 3.3M，所以选用 4M 的闪存（flash）作为该系统的非易失性存储器，且该系统的 flsah 被完整

刷新一遍大约需要 118 秒。

本实施例所选用 flsah 的最小单位即写入操作的最小基本操作块为 32K，将该 flash 划分为 4M/32K，即 128 个基本操作块，并将刷新过程的数据记录保存在该 flash 物理位置上的最后一个基本操作块中，数据记录中的 update_number 的取值范围为 0~127。该系统出厂时的 update_number 置为 0，update_flag 置为“Y”。

该系统进行刷新操作时，先读入刷新文件，判断 update_flag 的值为“Y”后，将 update_number 的值置为 0，将 update_flag 的值置为“N”，再开始刷新操作，即擦除一个基本操作块中的数据，再将相应的数据写入该基本操作块中，再擦除一个基本操作块中的数据，再将相应的数据写入该基本操作块中……，直到全部刷新完毕，将 update_flag 的值置为“Y”后结束。

如果该系统的刷新操作执行了 100 秒左右时突然发生意外断电的情况，而此时系统中 update_number 的值为 93，则表示本次刷新过程正在进行第 94（因为是从 0 开始计数，所以 93 表示第 94 块）块数据的写入时发生了中断。当系统重新进行刷新操作时，首先读取 flash 的最后一个基本操作块所记录的数据，由于上次刷新过程出现意外中断，因此得到 update_flag 的值不为“Y”，且 update_number 的值为 93，这表示系统的上次刷新过程没有正常结束，且刷新到第 94 块时中断。根据这些信息，刷新文件偏移到 93 × 32K，即 3047424 字节处继续进行刷新操作，直到将剩余的基本操作块全部刷新完毕，将 update_flag 的值置为“Y”后结束。这次刷新过程大约用时 21 秒完成。这样，既避免对已经写入成功的 flash 块进行重新的写入，又减少了刷新过程的时间，而且避免了再次发生意外的几率，间接地提高了刷新过程的安全性。

以上所述仅为本实用新型的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本发明的保护范围之内。

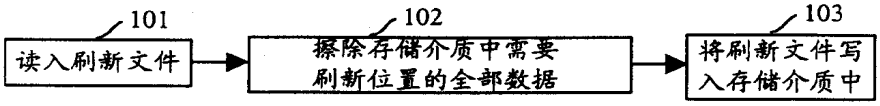

101
读入刷新文件
102
擦除存储介质中需要刷新位置的全部数据
103
将刷新文件写入存储介质中

图 1

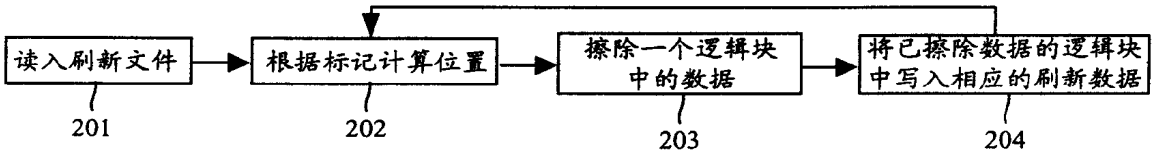

读入刷新文件
201
根据标记计算位置
202
擦除一个逻辑块中的数据
203
将已擦除数据的逻辑块中写入相应的刷新数据
204

图 2

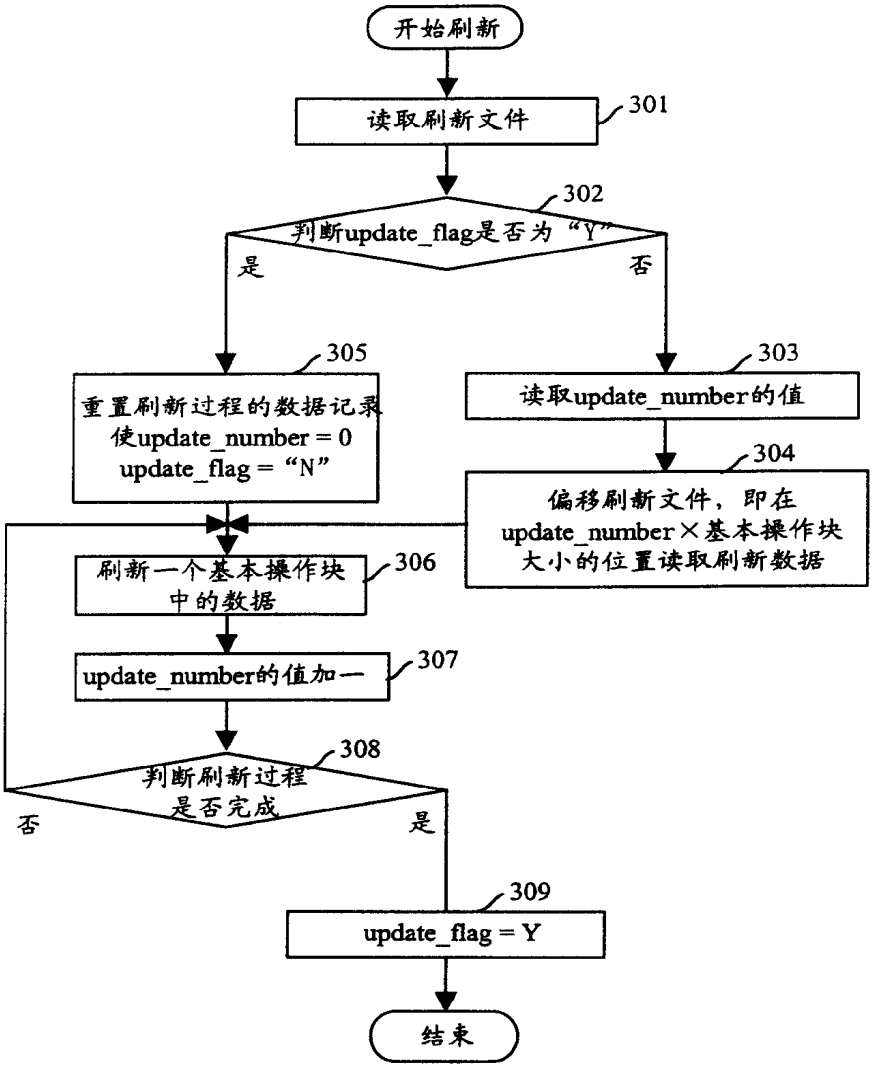

开始刷新
读取刷新文件
301
判断update_flag是否为“Y”
302
是
否
重置刷新过程的数据记录
使update_number = 0
update_flag = “N”
305
读取update_number的值
303
偏移刷新文件，即在
update_number×基本操作块
大小的位置读取刷新数据
304
刷新一个基本操作块
中的数据
306
update_number的值加一
307
判断刷新过程
是否完成
308
否
是
update_flag = Y
309
结束

图 3